WWW.SCIAM-MAGAZINE.NAROD.RU

Занимательный компьютер

Получение изображений самых сложных математических объектов с помощью компьютера-микроскопа

А. К. ДЬЮДНИ

В ЦЕНТРЕ огромного двумерного пространства чисел, называемого комплексной плоскостью, в молчаливой задумчивости пребывает множество Мандельброта. Если к числам комплексной плоскости многократно применять определенную математическую операцию, то числа, расположенные вне множества Мандельброта, стремятся к бесконечности. Числа, находящиеся внутри границ множества, остаются в его пределах, совершая иногда плавное, иногда колебательное движение. Область, где начинает проявляться неустойчивость, непосредственно примыкает к границе множества и отмечена особенно причудливыми траекториями. Здесь мы наблюдаем бесконечную регрессию форм, поражающую своей сложностью и необычной красотой.

Описанное множество носит имя Бенуа Мандельброта, члена научного общества фирмы IBM, работающего в Исследовательском центре Томаса Уотсона той же фирмы в Йорктаун-Хайтсе (шт. Нью-Йорк). Изучая геометрические формы, Мандельброт разработал теорию, названную им фрактальной геометрией. Это математическая теория форм, имеющих дробную размерность. Таким свойством обладает, в частности, граница множества Мандельброта, но она обладает также и другими интересными свойствами.

Используя сравнительно несложную программу, компьютер можно превратить в своеобразный микроскоп и наблюдать с его помощью за поведением границы множества Мандельброта. Теоретически можно «разглядывать» любую часть множества при любом увеличении (см. фотографию на обложке журнала и рисунки на с. 81 — 83). Издали множество напоминает приземистую, покрытую бородавками и лежащую на боку восьмерку. Внутренняя часть восьмерки зловеще черная, ее окружает яркая белая корона, которая постепенно темнеет, и на большом удалении от множества плоскость приобретает все более густой синий цвет, переходящий в черный.

При большем увеличении мы обнаруживаем, что каждая бородавка представляет собой крошечную фигурку, очень похожую на само множество. Перенесясь еще ближе к плоскости, чтобы лучше разглядеть одну из этих крошечных фигурок, мы видим совершенно другую картину: перед нами возникают ряды и спирали из бесчисленных усиков и причудливых завитушек, напоминающих биологические образования. Увеличив теперь одну из завитушек, мы видим, что она состоит из парных завитков, соединенных между собой тончайшим филигранным узором — мостиком. Увеличенное изображение мостика позволяет увидеть, что из его центра растут две завитушки. В центре же его центра, если можно так выразиться, находится четырехрядный мостик с четырьмя завитушками, а в центре этих завитушек мы обнаруживаем еще одну микроскопическую разновидность множества Мандельброта.

При еще большем увеличении выясняется, что эта версия множества Мандельброта отличается от исходной, хотя они и очень похожи. По мере дальнейшего увеличения в поле нашего зрения появляются уменьшенные версии уже встречавшихся форм, однако при внимательном изучении обнаруживаются какие-то различия. Этот процесс может продолжаться до бесконечности, и всюду мы видим бесконечное разнообразие форм, иногда даже поражающих своей красотой и великолепием.

В ЭТОЙ статье мы познакомимся с двумя программами, позволяющими исследовать результаты итерационных процессов (повторного применения операций), подобных тому, который приводит к образованию множества Мандельброта. С помощью первой программы были получены цветные картинки, приведенные в настоящей статье. При некоторых модификациях эту программу можно выполнять на персональном компьютере, имеющем необходимое аппаратное и математическое обеспечение графики. Картинки будут вполне удовлетворительными даже на черно-белом дисплее. Вторая программа предназначена для читателей, которым, как и мне, иногда хочется уйти от сложностей, возникающих в связи с понятием бесконечного, и ограничиться очевидной простотой конечного.

Слово «сложное» применяется здесь в двух смысловых значениях. Первое и обычное его значение вполне применимо при описании такого сложного объекта, как множество Мандельброта. Однако у этого слова есть другое, более формальное значение (на английском языке слово complex означает «сложный», формальное значение этого слова — «комплексный». — *Перев.*). Число называется сложное, или комплексное, когда оно состоит из двух частей, исторически сложившиеся названия которых — действительная и мнимая. Теперь эти термины уже не имеют особого значения — с таким же успехом две части комплексного числа могли бы называться толстой и тонкой. Так, например, $7 + 4i$ — это комплексное число, у которого действительная часть (толстая) равна 7, а мнимая (тонкая) равна $4i$. Буква i, стоящая при цифре 4, показывает, что эта часть является мнимой.

Каждое комплексное число можно представить точкой на плоскости. Плоскость комплексных чисел называется комплексной плоскостью. Чтобы найти положение числа $7 + 4i$ в комплексной плоскости, сначала найдем комплексное число 0, или $0 + 0i$, и отмерим семь единиц вправо и четыре единицы вверх. Точка с этими координатами представляет число $7 + 4i$. Комплексная плоскость содержит бесконечное количество таких точек — чисел. Их действительные и мнимые части могут быть либо положительными, либо отрицательными и быть либо целыми, либо дробными десятичными числами.

Комплексные числа легко складывать и умножать. Чтобы сложить числа $3 - 2i$ и $7 + 4i$, сложим их действительные и мнимые части отдельно. В результате получим сумму $10 + 2i$. Умножать комплексные числа лишь немногим сложнее. Если рассматривать символ i, например, как x в школьной алгебре, то произведение чисел $3 - 2i$ и $7 + 4i$ будет равно $21 + 12i - 14i - 8i^2$. Здесь уже следует пояснить, что $i^2 = -1$. Тогда полученное произведение можно упростить, суммировав его действительные и мнимые члены. Результатом умножения будет число $29 - 2i$.

Теперь можно описать итерационный процесс, порождающий множество Мандельброта. Начнем с алгебраического выражения $z^2 + c$, где z — комплексное число, которое может изменять свое значение (другими словами, z — это комплексная переменная), а c — некое постоянное ком-

WWW.SCIAM-MAGAZINE.NAROD.RU

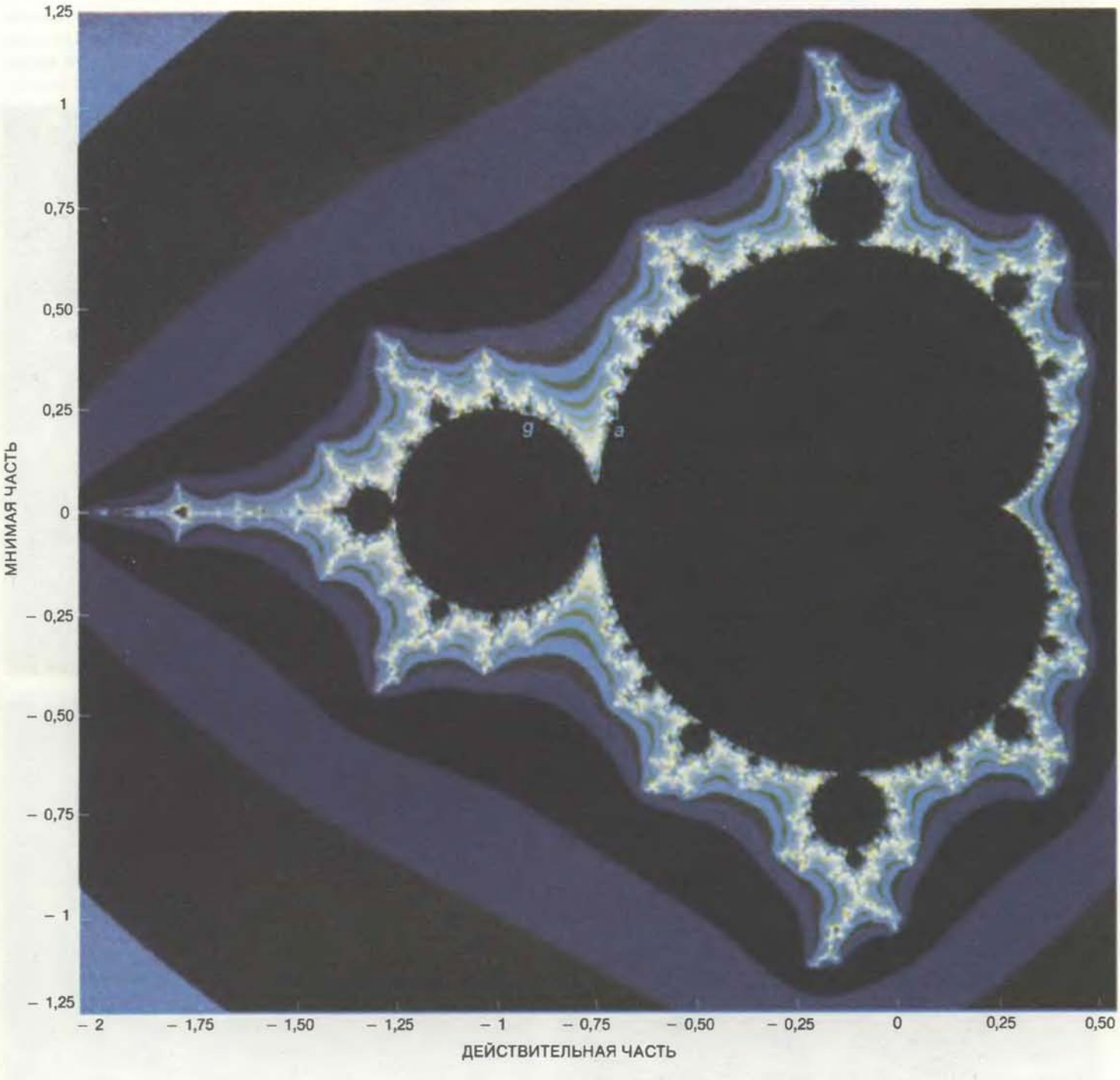
плексное число. Предположим вначале, что переменная z равна комплексному числу 0. Тогда квадрат z также равен 0, и, прибавив к z^2 константу c, получим c. Теперь подставим этот результат вместо z в выражении $z^2 + c$. Получим новую сумму $c^2 + c$. Опять подставим ее вместо z. Следующая сумма будет равна $(c^2 + c)^2 + c$. Будем продолжать этот процесс, используя выходные данные предыдущего шага в качестве входных данных следующего.

Применяя описанный итерационный процесс для каких-то определенных значений c, мы наблюдаем довольно странную картину. Вот, например, что получается, когда c равно $1 + i$:

первая итерация, $1 + 3i$,
вторая итерация, $-7 + 7i$,
третья итерация, $1 - 97i$.

Отметим, что действительная и мнимая части могут расти, уменьшаться или менять знак. Если этот процесс продолжать, то в результате будем получать все бóльшие комплексные числа.

А что следует понимать под величиной комплексного числа? Поскольку комплексные числа соответствуют точкам на плоскости, напрашивается мысль воспользоваться в качестве критерия расстоянием. Величина комплексного числа — это как раз расстояние от представляющей его точки до точки 0. Геометрически это расстояние выражается гипотенузой прямоугольного треугольника, катеты которого образованы действительной и мнимой частями комплексного числа. Поэтому, чтобы найти величину комплексного числа, нужно возвести в квадрат его действительную часть и коэффициент при мнимой части, сложить их и извлечь из суммы квадратный корень. Например, величина комплексного числа $7 + 4i$ равна квадратному корню из $7^2 + 4^2$, или приблизительно 8,062. Когда в результате многократного применения описанного выше итерационного процесса комплексные числа достигают определенной величины, они начинают рас-



Множество Мандельброта на комплексной плоскости. На диаграмме выделены детали, показанные на обложке журнала и на следующих трех рисунках

WWW.SCIAM-MAGAZINE.NAROD.RU

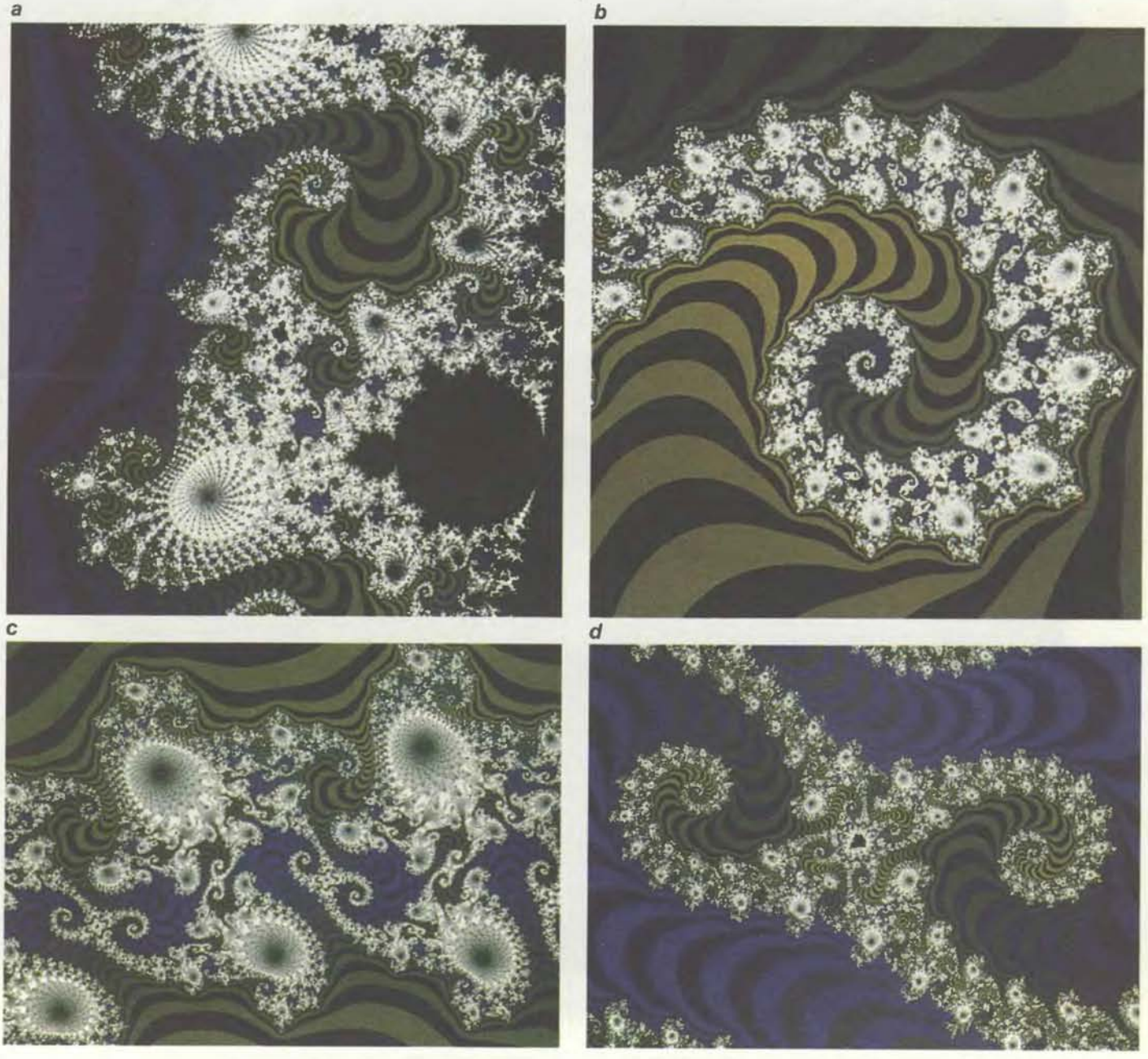
ти очень быстро. И в самом деле, уже через несколько следующих шагов итерации они превышают максимальное число, которое можно записать в вычислительной машине.

К счастью, мы можем игнорировать все комплексные числа c, которые стремительно уносятся в бесконечность. Множество Мандельброта — это множество всех комплексных чисел c, для которых величина выражения $z^2 + c$ остается конечной даже после бесконечно большого количества шагов итерации. Программа, которую я собираюсь описать, как раз и отыскивает такие числа. Имеющейся у меня информацией по этому вопросу я обязан Дж. Хаббарду, математику из Корнеллского университета. Хаббард — специалист по множеству Мандельброта, он один из тех, кто первым с помощью вычислительной машины получил его графические изображения. Бóльшая часть изображений, приведенных в настоящей статье, были получены Х. Пайтгеном и его коллегами из Бременского университета. Однако этому искусству Пайтген научился у Хаббарда.

ПРОГРАММА, которую я называю MANDELZOOM, была создана по образу и подобию программы Хаббарда. В программе организуется массив *pic*, необходимый для хранения картинок. Элементы массива *pic* представляют собой отдельные элементы картинки, называемые пикселами. Они расположены в форме решетки. В массиве программы Хаббарда насчитывалось 400 столбцов и 400 строк, а в программе Пайтгена еще больше. Читатели, желающие приспособить программу MANDELZOOM для своего персонального компьютера, могут сами выбрать размеры массива в зависимости от типа имеющегося в их распоряжении компьютера. Чем больше массив, тем больше время ожидания картинки, но зато тем лучше разрешающая способность изображения.

На начальном этапе работы программы необходимо выбрать квадратную область комплексной плоскости, которую вы хотели бы рассмотреть. Для этого нужно указать комплексное число, соответствующее левому нижнему углу выбранного квадрата. Наличие двух переменных в программе, *acorner* и *bcorner*, позволяет вводить соответственно действительную и мнимую части этого числа. Затем нужно указать длину стороны квадрата, присвоив соответствующее значение переменной *side*.

В следующей части программы происходит подгонка массива *pic* под размеры выбранного квадрата путем



Последовательные увеличения области **a**, *отмеченной на предыдущем рисунке*

WWW.SCIAM-MAGAZINE.NAROD.RU

вычисления значения переменной *gap*. *Gap* — это расстояние между двумя соседними пикселами в выбранном для рассмотрения квадрате. Чтобы получить *gap*, поделим значение *side* на число строк (или столбцов) в массиве *pic*.

Третья часть программы — центральная. Она отыскивает комплексные числа *c*, принадлежащие множеству Мандельброта, а также присваивает цвета числам, расположенным, в некотором специальном смысле, неподалеку. Процедура выполняется для каждого отдельного пиксела. Таким образом, в программе Хаббарда, имеющей массив размером 400 × 400, потребуется 160 000 шагов вычисления. Предположим, что в данный момент программа обрабатывает данные, относящиеся к пикселу, принадлежащему строке *m* и и столбцу *n*. Тогда процедура, выполняемая третьей частью программы, распадается на четыре этапа, или шага.

1. Вычислить комплексное число *c*, соответствующее данному пикселу: сложить величину *n* × *gap* с *acorner*, чтобы получить действительную часть *ac* числа *c*; сложить величину *m* × *gap* с *bcorner*, чтобы получить мнимую часть *bc* числа *c*. При этом нет необходимости включать в программу символ *i*.

2. Вначале задать комплексную переменную *z* (с действительной частью *az* и мнимой частью *bz*) равной 0 + 0*i*. Установить целую переменную *count* равной 0.

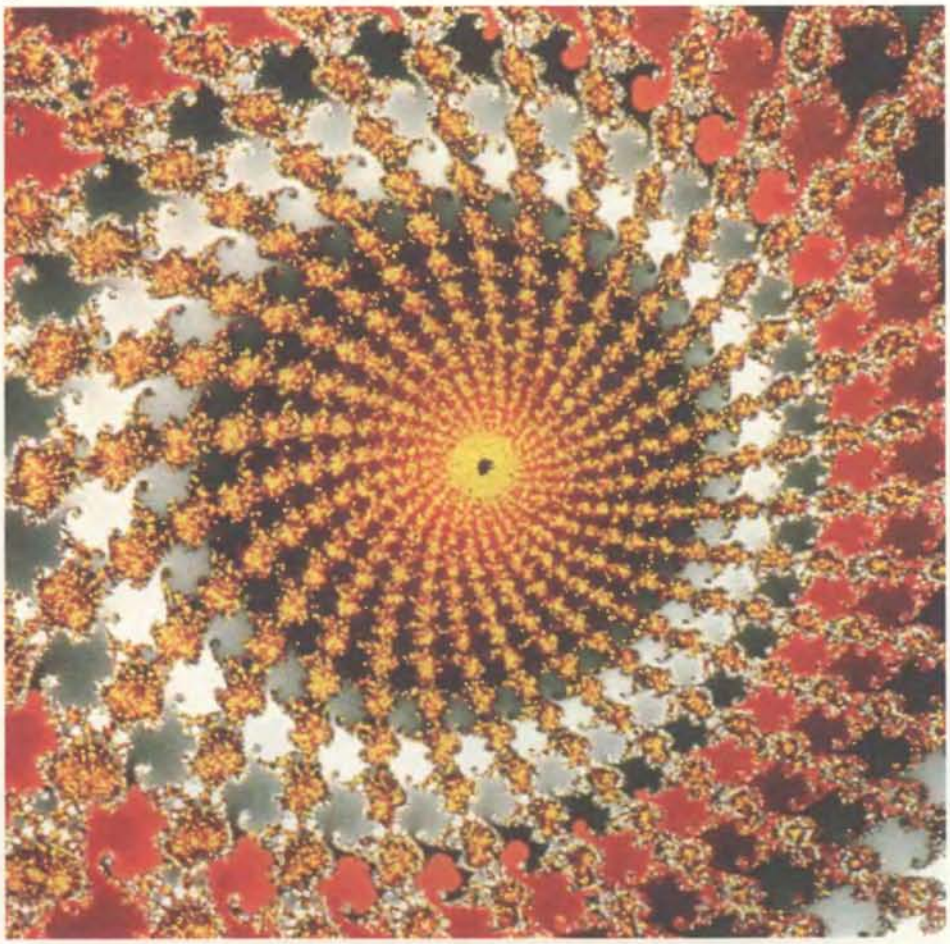
3. Выполнять повторно следующие три операции, пока или величина *z* не превысит 2, или значение *count* не превысит 1000, независимо от того, какое из этих событий наступит раньше:

$z \leftarrow z^2 + c$
count ← *count* + 1
side ← ⟨величина *z*⟩

Почему цифра 2 играет такую важную роль? Дело в том, что, согласно теории, комплексное число *z*, к которому применяется итерационный процесс, «уйдет» в бесконечность тогда и только тогда, когда на определенной стадии процесса оно достигнет значения 2 или более. Как показывают вычисления, довольно много точек, которым суждено уйти в бесконечность, достигают 2 уже после первых нескольких шагов итераций. Более «медленные» точки достигают этой величины все реже и реже по мере роста значения переменной *count*.

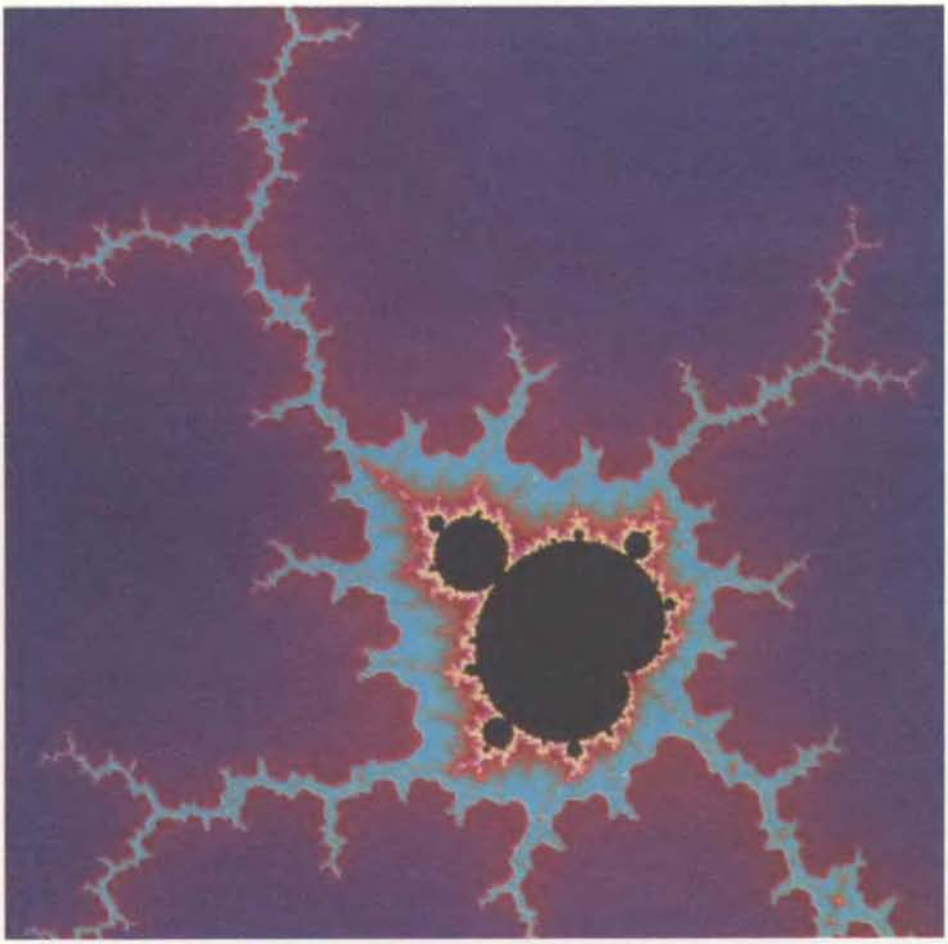
4. Присвоить цвет элементу *pic* (*m*, *n*) в соответствии со значением переменной *count*, достигнутым после выполнения 3-го шага процедуры. Высветить цвет данного пиксела на экране дисплея. Заметим, что цвет пиксе-

e



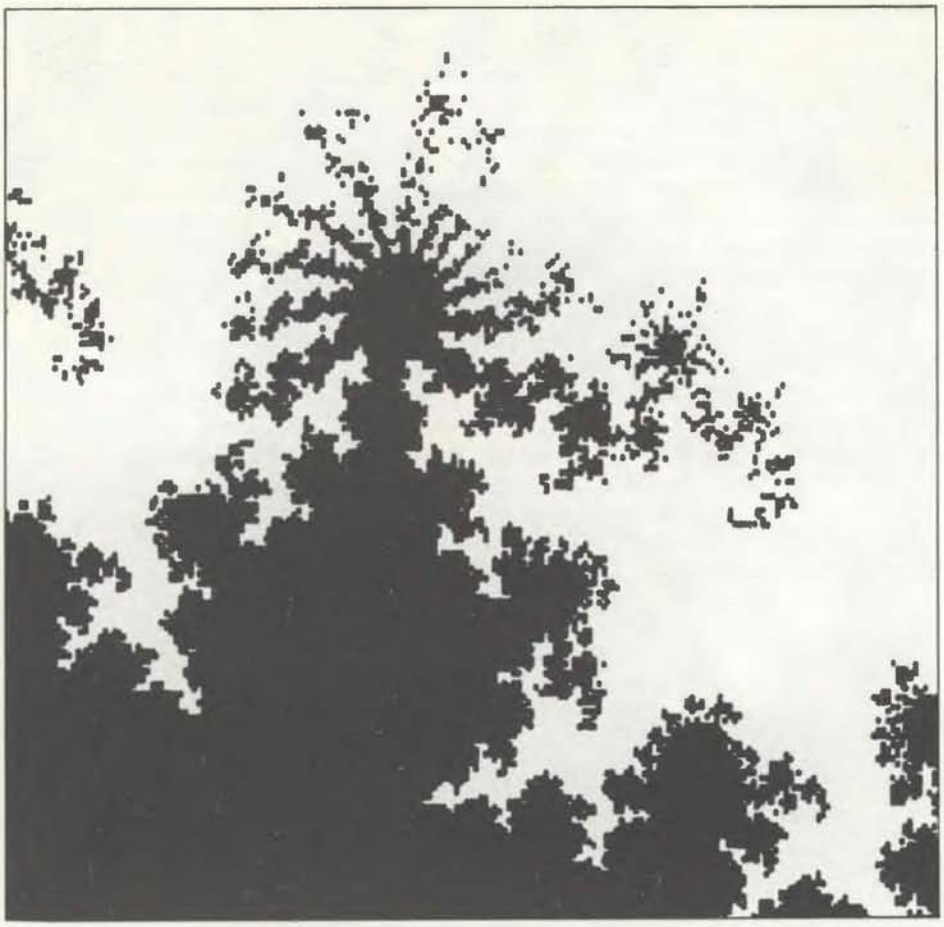
Зрачок глаза, который смотрит из центра области, изображенной на предыдущем рисунке в кадре d

f



Миниатюрное множество Мандельброта в области f *диаграммы на с. 81, связанное с основным множеством тонкой нитью*

WWW.SCIAM-MAGAZINE.NARODRU



Точечное изображение миниатюрного множества Мандельброта, полученное на черно-белом дисплее

ла зависит только от одного комплексного числа, находящегося в его ближайшей окрестности, а именно от числа, находящегося в верхнем правом углу этой окрестности. Поведение этого числа определяет поведение всего пиксела.

При таком способе присвоения цвета нужно, чтобы диапазон изменения переменной *count* разделялся на поддиапазоны, по одному на каждый цвет. Пикселы, для которых величина z достигает 2 уже после первых нескольких итераций, окрашиваются в красный цвет. Пикселы, для которых величина z достигает 2 после сравнительно большого количества итераций, окрашиваются в фиолетовый цвет, расположенный на противоположном краю спектра. Пикселы, для которых величина z остается меньше 2 даже после 1000 итераций, по предположению, лежат в множестве Мандельброта и окрашиваются в черный цвет.

Было бы разумно оставлять цвета неопределенными, пока не будет установлен диапазон изменения переменной *count* в данном конкретном квадрате. Если диапазон оказывается узким, то весь цветовой спектр можно уместить в его пределах. Поэтому Хаббард предлагает, чтобы на 4-м шаге процедуры каждому элементу массива *pic* присваивалось лишь значение переменной *count*. Затем отдельная программа должна будет просмотреть массив, определить наибольшее и наименьшее значения *count* и соответствующим образом распределить спектр. Читатели, которым удастся написать все остальные фрагменты программы, несомненно, справятся и с этим местом.

При отсутствии цветного дисплея можно обойтись и черно-белым. Комплексные числа, величина которых превышает 2 после r итераций, окрашиваются в белый цвет, остальные точки остаются черными. Значение параметра r можно выбрать по вкусу. Чтобы не просиживать целую ночь перед экраном в ожидании результатов, размер массива можно, например, принять равным 100×100. Хаббард считает также, что разумно было бы сократить максимальное количество итераций с 1000 до 100. Результатом подобной программы будет точечный образ, или черно-белое представление цветного изображения, о котором говорилось выше (см. рисунок вверху).

КАКОВА же мощность микроскопа, реализованного на персональном компьютере? В какой-то степени она зависит от количества разрядов чисел, которыми может оперировать машина. Например, в персональном компьютере фирмы IBM используется микропроцессор 8088 фирмы Intel Corporation, предназначенный для выполнения операций над двоичными числами длиной 16 разрядов. Пользуясь так называемой двойной точностью, можно увеличить количество разрядов до 32. При этом условии, по моим подсчетам, можно достигнуть увеличения порядка 100 000. Используя специальное программное обеспечение, объединяющее двоичные разряды чисел в единые цепочки, можно повысить точность вычислений до сотен значащих двоичных разрядов. При такой точности теоретически достижимое увеличение множества Мандельброта намного превышает увеличение, необходимое для того, чтобы рассмотреть внутреннюю структуру атомного ядра.

Какую часть комплексной плоскости интереснее исследовать? Конечно, в окрестности множества Мандельброта. Ну а точнее? По словам Хаббарда, на плоскости «есть зиллионы прекраснейших мест». Подобно туристу, посетившему необычайно красивые места в мире бесконечного, он может буквально засыпать читателей предложениями тех мест, какие будут особенно интересными. Правда, эти места не носят таких экзотических названий, как Сингапур или Таити. Хаббард рекомендует читателям «исследовать область комплексных чисел с диапазоном действительных частей от 0,26 до 0,27 и мнимых — от 0 до 0,01». Он также предложил две другие области:
Действительная часть

от $-0{,}76$ до $-0{,}74$
от $-1{,}26$ до $-1{,}24$

Мнимая часть

от 0,01 до 0,03
от 0,01 до 0,03

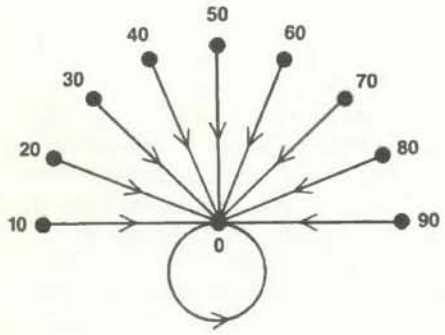
Рассматривая цветные картинки, иллюстрирующие нашу статью, следует иметь в виду, что точки, окрашенные в любой другой цвет, кроме черного, не принадлежат множеству Мандельброта. В значительной мере красота картинок обусловлена гаммой красок, ассоциируемых с точками, устремляющимися в бесконечность. На самом деле если бы мы захотели посмотреть на множество Мандельброта в полной изоляции от окружающих его точек, то картинка не была бы столь красивой: мы увидели бы перед собой множество, всюду покрытое тонкими волосками и содержащее миниатюрные структуры, очень похожие на само множество.

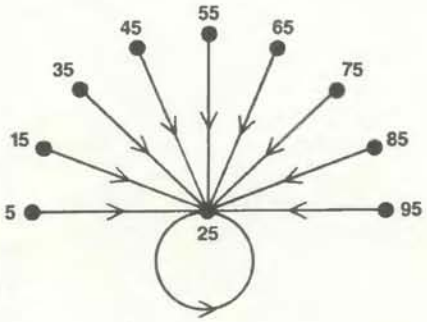
В действительности ни одно из миниатюрных множеств Мандельброта не является точной копией родительского множества. Более того, между этими миниатюрными версиями мно-

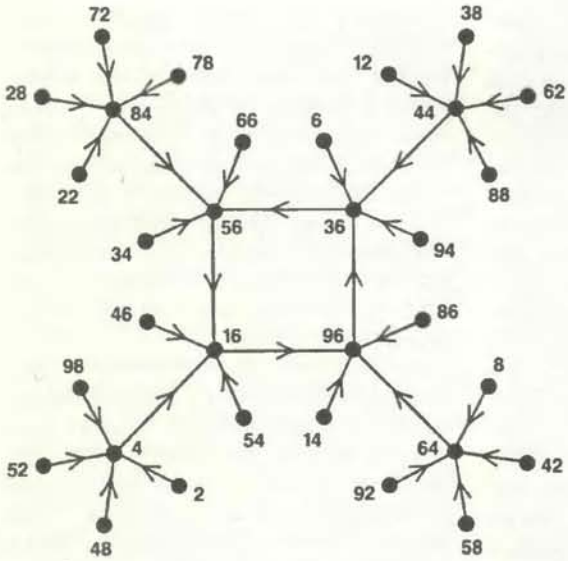
WWW.SCIAM-MAGAZINE.NAROD.RU

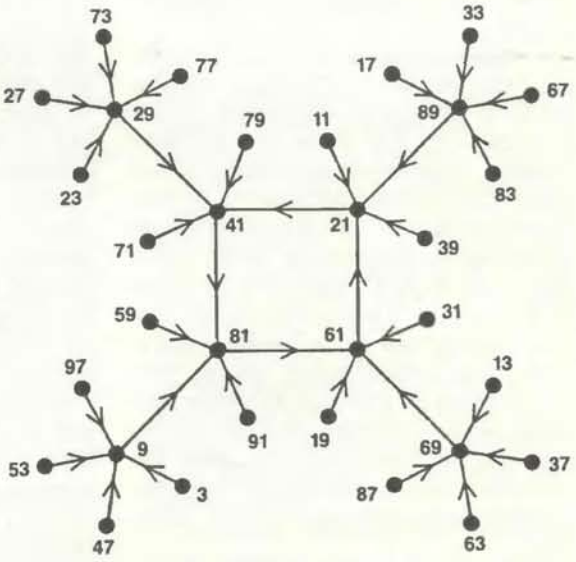
жества также имеются некоторые различия. В окрестности множества Мандельброта мы видим многочисленные его микроскопические версии, как бы подвешенные в «атмосфере» комплексной плоскости. Однако это впечатление обманчиво. Согласно удивительной теореме, доказанной Хаббардом и его коллегой А. Дуади из Парижского университета, множество Мандельброта является связным. Поэтому даже миниатюрные версии множества, кажущиеся изолированными в комплексной плоскости, на самом деле соединены с родительским множеством тонкими волокнами. Миниатюрные версии множества встречаются повсюду в окрестности и имеют самые разные размеры. В каждом квадрате конечного размера содержится бесконечное количество миниатюрных версий множества, из которых при любом увеличении можно разглядеть лишь несколько. По словам Хаббарда, множество Мандельброта представляет собой «самый сложный объект в математике».

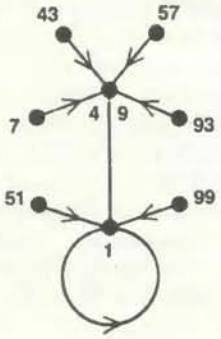
ПОСЛЕ путешествия в мир бесконечно сложного приятно отдохнуть в пределах конечного. Многократно повторяя операцию возведения в квадрат на конечном множестве обыкно-

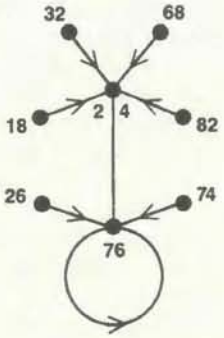












Шесть компонент диаграммы итерационного процесса возведения в квадрат для первых 100 натуральных чисел

WWW.SCIAM-MAGAZINE.NAROD.RU

венных целых чисел, можно также получить интересные структуры. Правда, эти структуры не геометрические, а, скорее, комбинаторные.

Выберем наобум любое целое число в диапазоне от 0 до 99. Возведем его в квадрат и оставим два последних десятичных разряда. Они также будут представлять собой число в диапазоне от 0 до 99. Например, возведя 59 в квадрат, мы получим 3481; два последних разряда этого числа 81. Если продолжить этот процесс, то рано или поздно мы получим число, уже встречавшееся нам ранее. Например, начиная с числа 81, мы получим последовательность 61, 21, 41 и 81, и эта последовательность из четырех чисел будет теперь повторяться до бесконечности. Оказывается, что такие циклы всегда возникают при итерационных процессах над конечными множествами. Нетрудно видеть, что существует по крайней мере одно повторяющееся число после 100 операций на множестве из 100 чисел. Первое же повторяющееся число приводит к возникновению цикла. Существует изящная программа обнаружения подобных циклов, почти не требующая памяти, но об этом несколько позже.

Приблизительно за час можно построить диаграмму результатов описанного процесса. Представим каждое число от 0 до 99 точкой на листе бумаги. Если после возведения в квадрат одно число переходит в другое, то соединим соответствующие точки стрелочкой. Например, такую стрелочку мы должны провести от точки 59 к точке 81. После нескольких первых соединений на диаграмме могут возникнуть запутанные циклы, поэтому время от времени их следует перерисовывать таким образом, чтобы стрелки никогда не пересекались. Нарисовать такую диаграмму без пересечений, оказывается, всегда возможно.

Можно пойти даже дальше. В ходе процесса часто возникают отдельные поддиаграммы, и их можно построить таким образом, чтобы выявились некоторые свойства симметрий, характерных для итераций. Например, непересекающаяся диаграмма итераций для возведения в квадрат целых чисел в диапазоне от 0 до 99 содержит шесть не связанных между собой диаграмм. Части диаграммы встречаются в виде идентичных пар, и каждая часть обладает высокой степенью симметрии (см. рисунок на с. 85). Может ли читатель дать толкование этих симметрий? Что изменится, если диапазон расширить от 0 до 119? Существует ли какая-нибудь зависимость между числом не связанных друг с другом частей на диаграмме и наибольшим целым числом в последовательности?

Сходные с только что описанными закономерности наблюдаются и при итерациях, применяемых к некоторым комплексным числам, принадлежащим множеству Мандельброта: для определенных значений c итерации, применяемые к выражению $z^2 + c$, могут привести к конечному циклу из комплексных чисел. Так, например, начиная с комплексного числа $0 + 1i$, мы получаем бесконечные колебания между двумя числами: $-1 + 1i$ и $0 - 1i$. Цикл может даже состоять из единственного числа. Такие циклы независимо от того, встречаются они на конечном множестве или на бесконечном множестве Мандельброта, называются аттракторами.

Каждая из шести частей диаграммы итерационного процесса для целых чисел в диапазоне от 0 до 99 содержит один аттрактор. Геометрически аттрактор можно представить в виде многоугольника, а множества чисел, которые приводят к нему, можно представить в виде деревьев.

Один из возможных способов найти аттрактор заключается в том, чтобы запоминать каждое порождаемое число в специально отведенном массиве программы. Только что полученное число следует сравнить с каждым из ранее запомненных в массиве чисел. Если среди них находится число, совпадающее с данным, то все числа, расположенные между найденным числом и только что полученным, равным ему, выводятся на экран или печать. Способ довольно простой, и его нетрудно реализовать в программе. Однако, если массив большой, процесс сравнения чисел может потребовать много времени. Чтобы найти аттрактор в массиве, содержащем n чисел, необходимо выполнить порядка n^2 операций сравнения, поскольку каждое новое число нужно сравнить с каждым числом массива, а количество последних может доходить до n.

Существует маленькая остроумная программа, которая находит аттракторы значительно быстрее. Вместо n слов памяти программа требует лишь два, и ее можно составить даже на простейшем карманном программируемом калькуляторе. С этой программой вы сможете познакомиться, прочтя любопытную книжку: Hoffman D., Mohler L. Mathematical Recreations for the Programmable Calculator («Занимательная математика для программируемого калькулятора»). Разумеется, многие рассмотренные в книжке программы могут быть легко переписаны для персонального микрокомпьютера или любой другой ЭВМ.

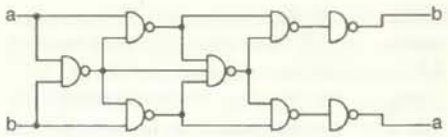


Схема пересечения сигналов из 8 вентилей И-НЕ

Программа RHOP получает последовательность чисел, которая в конце концов зацикливается и напоминает кусок каната с петлей на одном конце. Она напоминает также греческую букву «ро» (ρ) (по-английски rho; отсюда название программы RHOP. — *Перев.*). В программе две переменные — *fast* и *slow* (соответственно «быстрая» и «медленная». — *Перев.*). Первоначально обеим переменным присваивается значение исходного числа. Итерационный цикл программы состоит лишь из трех команд:

$$fast \leftarrow fast \times fast \pmod{100}$$
$$fast \leftarrow fast \times fast \pmod{100}$$
$$slow \leftarrow slow \times slow \pmod{100}$$

Операция «mod 100» извлекает последние два разряда произведения. Заметим, что число *fast* дважды возводится в квадрат, а число *slow* только один раз. Число *fast* пробегает путь от хвоста буквы «ро» до ее головы вдвое быстрее числа *slow*. Уже в пределах головы буквы «ро» число *fast* встречается с числом *slow*. Программа останавливает итерационный цикл, когда совпадают значения переменных *fast* и *slow*.

Аттрактор определяется путем повторного возведения в квадрат числа, являющегося на данный момент значением переменной *slow*. Когда это число встречается вновь, программа останавливается и высвечивает или распечатывает последовательность чисел между двумя одинаковыми значениями *slow*.

Читатели могут попытаться построить диаграммы на основе результатов итерационного процесса возведения в квадрат чисел из конечных множеств различных размеров. Эти диаграммы можно получить как с помощью компьютера, так и от руки. Дискретные итерационные процессы — это новая, развивающаяся область математики, которая находит применение в вычислительной технике, информатике, биоматематике, физике и даже социологии. Теоретики, наверняка, с интересом прочтут монографию Франсуа Роберта из Гренобльского университета, которая посвящена этому предмету и в настоящее время готовится к печати.

WWW.SCIAM-MAGAZINE.NAROD.RU

ДВУМЕРНЫЕ существа, населяющие планету Арда, выражают глубокую признательность многочисленным читателям, попытавшимся усовершенствовать логическую схему с пересечением сигналов, описанную в июльском 1985 г. номере журнала «В мире науки». Эта схема состоит из 12 вентилей И-НЕ с двумя входами. Я обратился к читателям с просьбой найти минимальное количество вентилей И-НЕ (и только вентилей И-НЕ), необходимых для построения такой схемы. Большинство представленных решений содержит 10 вентилей, что, безусловно, является некоторым улучшением, однако трем читателям удалось построить схему пересечения, состоящую из восьми вентилей (см. рисунок на с. 86).

В схеме, состоящей из восьми вентилей, есть один вентиль И-НЕ с тремя входами и два вентиля И-НЕ с одним-единственным входом. Последние действуют как инверторы, преобразующие 0 на входе в 1 на выходе и наоборот. Схему с восемью вентилями удалось построить Э. Карлсону из Кембриджа (шт. Массачусетс), Д. Кеппу из Сан-Хосе (шт. Калифорния) и С. Салливану из Бевертона (шт. Орегон). Я сообщил их имена вместе с предложенными ими вариантами усовершенствованной схемы пересечения моим друзьям ардеанам. Хотите верьте, хотите нет, но точно такая же схема пересечения из восьми вентилей была запатентована в США 26 апреля 1966 г. Номер патента 3 248 573. Р. Фрэнк, консультант по вычислительной технике из Бирмингема (шт. Мичиган), сообщил, что этот патент принадлежит Л. Спэндорферу из Челтенхема (шт. Пенсильвания), А. Тонику из Дрешера (шт. Пенсильвания) и Ш. Ивену из Кембриджа (шт. Массачусетс).

Естественно, возникает вопрос, применяется ли эта схема в каком-либо современном электронном устройстве? Возникает также вопрос о том, существует ли схема пересечения сигналов, содержащая еще меньшее число вентилей И-НЕ. По всей видимости, нет.

В письме У. Джонсона из Лонг-Бича (шт. Калифорния) описаны разнообразные плоские схемы, построенные из вентилей различных типов. Оказывается, можно построить не только двумерные схемы пересечения сигналов, но также и плоские триггеры. Напомним, что на триггерах основаны элементы памяти двумерного компьютера.

Одномерные компьютеры в виде клеточных автоматов изучались С. Уолфрэмом из Принстонского научно-исследовательского института. Пока еще слишком рано говорить о том, какой вклад в эту науку внесли читатели, ознакомившись с брошюрой Уолфрэма, в которой описаны катапульты, стреляющие планерами, однако письма читателей уже поступают, и можно подвести некоторые предварительные итоги. Во-первых, из-за ошибки, закравшейся в статью, посвященную одномерным автоматам, несколько читателей занялись исследованием автомата с кодовым номером 792. Уолфрэм и я имели в виду автомат 357. Во-вторых, я сожалею о том, что не упомянул об одномерном автомате, обладающем свойством универсальных вычислений. У меня была мысль описать такой автомат, впервые построенный А. Смитом в 1970 г. В то время Смит был аспирантом Станфордского университета. Но я боялся, что описание универсального одномерного автомата Смита чрезмерно усложнит статью, так как автомат обладает 18 состояниями ($k = 18$) и окрестностью из трех клеток ($r = 1$).

А. Рубин из Лос-Анджелеса внес разумное предложение определить скорость света (скорость распространения сигнала) в произвольном линейном автомате. Предложение Рубина устраняет дефект в определении, принятом ранее, согласно которому скорость света полагалась равной одной клетке в единицу времени. В старом определении не учитывалась возможность, что не все автоматы способны достигать такой скорости. Согласно новому определению, «скорость света — это максимальная скорость распространения любого импульса (скажем, вправо)». Передний фронт распространения импульса определяется условием, что справа от него могут быть только нули. Далее Рубин доказывает, что скорость света для автомата с кодовым номером 792 равна 1/3.

В июльском 1985 г. номере журнала «В мире науки» я также обратился к читателям с вопросом о том, обладает ли одномерный автомат «Волна» односторонней катапультой. Планеры, запущенные такой катапультой, должны двигаться вправо до бесконечности, и ни один из них никогда не должен двигаться влево. У. Б. Липп из Милфорда (шт. Коннектикут) привел простой и изящный аргумент против существования такой катапульты. «Рассмотрим конфигурацию, — пишет он, — никогда не имеющую ненулевых значений слева от некоторого блока, помеченного как 0. Заметим, что крайним слева ненулевым значением конфигурации должна всегда быть 1. Если бы оно было равно 2, то эта двойка начала бы двигаться влево до бесконечности, что привело бы к противоречию с предположением, согласно которому слева от блока 0 нет никаких ненулевых значений. Однако крайняя слева 1 на следующем цикле должна превратиться в 0, в результате чего левая граница конфигурации должна переместиться по меньшей мере на один блок вправо». Таким образом, мы приходим к выводу, что либо возникает планер, движущийся влево, либо катапульта разрушается наступающими слева нулями.

Другие читатели попытались доказать, что автомат «Волна» не обладает способностью универсальных вычислений. Для некоторых автоматов можно доказать достаточное условие, а именно что задача остановки является решаемой. «Волна» останавливается, когда во всех ее клетках нули, однако условия остановки для универсальной вычислительной машины, которую автомат может в себе заключить, вообще говоря, могут оказаться совершенно отличными.

Несколько читателей попытались сконструировать одномерный автомат, способный проводить универсальные вычисления. Среди них Ф. Адамс из Ист-Харфорда (шт. Коннектикут), Дж. Амстердам из Кембриджа (шт. Массачусетс), К. Игуса из Брандейского университета и К. Кейди из Ист-Пеории (шт. Иллинойс). Все присланные конструкции сравнительно несложны и на вид правдоподобны, однако Кейди, не удовлетворившись одномерным автоматом, предложил автомат нулевой размерности. Он должен состоять из одной-единственной клетки, и поэтому логично назвать эту клетку точечным автоматом. Читатели с теоретическими наклонностями, возможно, испытают удовольствие, размышляя о том, обладает ли точечный автомат способностью к универсальным вычислениям. Возможен ли такой автомат?

А. Смит опубликовал доказательство существования одномерного автомата, обладающего способностью к универсальным вычислениям, в 1971 г. в журнале «ACM» («Journal of the Association for Computing Machinery»). После такого многообещающего начала можно было подумать, что карьера Смита будет продолжена в каком-нибудь известном академическом центре. Однако талант Смита расцвел совершенно в другой области — в настоящее время он является директором лаборатории машинной графики в кинематографической фирме Lucasfilm, Ltd. в Сан-Рафаэле (шт. Калифорния). В одной из своих будущих статей я надеюсь рассказать вам о некоторых поразительных кинематографических эффектах, полученных в лаборатории фирмы Lucasfilm.